

- 1 是 wakeup channel, - 2 是 timerfd_channel

删除 1 条消息



在 Event 中的 loop 当中 每一次循环都即会检测到是否有事件发生
如果 poll 中有事件发生

每个 channel 的回调函数如何被调用
channel 是一个传输事件的数据

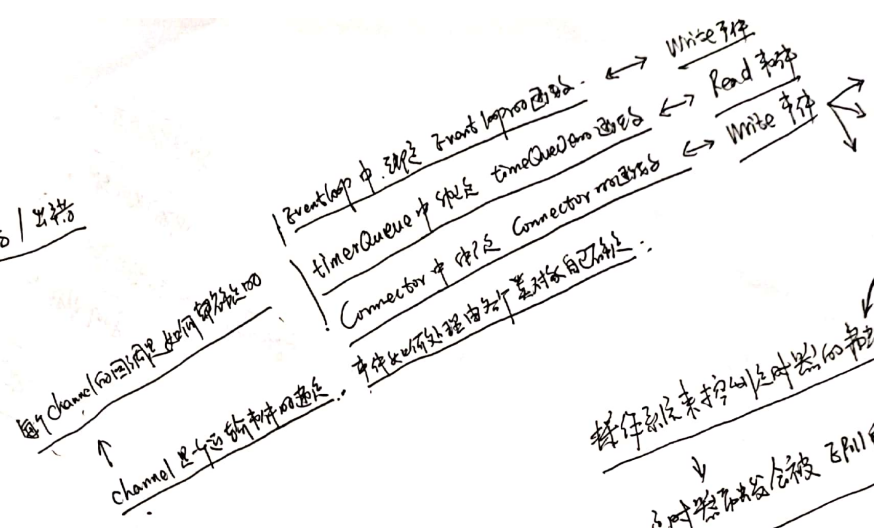
Eventloop 中 调起 Front loop 函数
timerQueue 中 调起 timerQueue 函数
Connector 中 调起 Connector methods
事件如何被处理由 类对象自己控制

事件被用来控制定时器的参数
定时器参数会被 Epoll 所监视到
timerfd_channel 的回调函数被调用 执行 handle Read



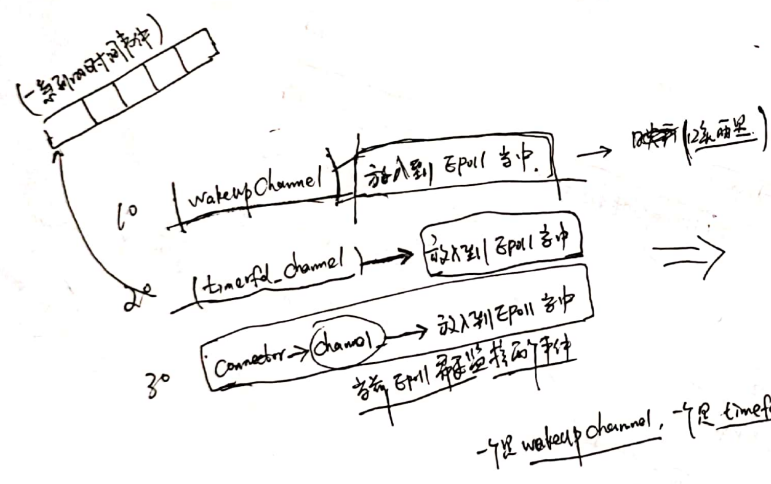
构建的过程

- ② 每当有事件发生的时候，都会将对应的channel有状态（在eventloop中进行设置）
- ④ 返回的对象的地址无解几部：读/写/出错



每个 channel 的回调函数如何被调用
channel 是子进程的回调函数

操作系统的定时器被触发
定时器被触发会被 epoll 所检测到。
timerfd-channel 的回调函数被调用。执行 handle Read。



在 Eventloop 中的 loop 当中 每次循环即会检测是否有事件发生
如果 poll 中有事件发生

一个 wakeupChannel, 一个 timerfd-channel

- 1. (epoll 监听的对象即有定时调度的 / 也有非定时调度的) / 所有放入到 epoll 中的回调函数，时间间隔... 都必须为非阻塞形式的
- 2. PendingFunction - 中有的回调函数来自于各个对象的回调 / 这些回调函数可能会生成一些事件并放入到 epoll 中 / 要求非阻塞。

